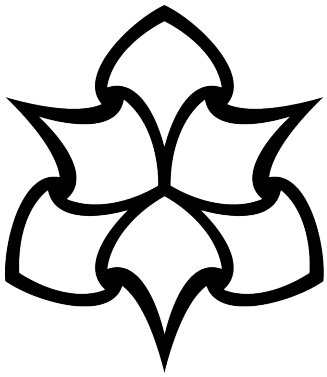


Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail[1]

Sushil Punia

Vishwakarma Bhawan
Dept. of Management Studies
Indian Institute of Technology Delhi
Hauz Khas, New Delhi 110016, India.
s.punia.official@gmail.com; sushil.punia@dms.iitd.ac.in

Konstantinos Nikolopoulos* (Corresponding Author)

forLAB, Bangor Business School
Bangor University
Hen Colleg 2.06
Gwynedd, Wales, LL57 2DG, UK.
k.nikolopoulos@bangor.ac.uk

Surya Prakash Singh

Vishwakarma Bhawan
Dept. of Management Studies
Indian Institute of Technology Delhi
Hauz Khas, New Delhi 110016, India.
surya.singh@gmail.com

Jitendra K. Madaan

Vishwakarma Bhawan
Dept. of Management Studies
Indian Institute of Technology Delhi
Hauz Khas, New Delhi 110016, India.
jmadaan@dms.iitd.ac.in

Konstantia Litsiou

Department of Marketing, Retail and Tourism
Manchester Metropolitan University Business School
All Saints Campus, Oxford Road, Manchester, M15 6BH, UK.
k.litsiou@mmu.ac.uk

[1] Accepted in International Journal of Production Research on 20 February 2020. PURE – Repository version for GREEN ACCESS.

Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail

Abstract

This paper proposes a novel forecasting method that combines the deep learning method - long short-term memory (LSTM) networks and random forest (RF). The proposed method can model complex relationships of both temporal and regression type which gives it an edge in accuracy over other forecasting methods. We evaluated the new method on a real-world multivariate dataset from a multi-channel retailer. We benchmark the forecasting performance of the new proposition against neural networks, multiple regression, ARIMAX, LSTM networks, and RF. We employed forecasting performance metrics to measure bias, accuracy, and variance, and the empirical evidence suggests that the new proposition is (statistically) significantly better. Furthermore, our method ranks the explanatory variables in terms of their relative importance. The empirical evaluations are replicated for longer forecasting horizons, and online and offline channels and the same conclusions hold; thus, advocating for the robustness of our forecasting proposition as well as the suitability in multi-channel retail demand forecasting.

Keywords: Deep learning; LSTM networks; Random forests; Multi-channel; Retail.

1. Introduction

Firms such as manufacturers, distributors, retailer, etc., are always in search of more accurate forecasts because that would lead to less uncertainty in decision-making. Particularly in retail, accurate demand forecasting leads to informed decisions in purchasing, inventory management, scheduling, capacity management, assortment planning, etc. In the last few years, a growing body of literature promoted a 'horses for courses' approach which advocates that different class of forecasting methods is expected to be more suitable for different types of data (Petropoulos et al. 2014). The most commonly used methods for demand forecasting are time-series methods that mainly try to identify trend and cyclicity in the series, and multivariate methods that establish relationships among the variable of interest and other independent variables. However, these methods struggle to perform when the dependent variable (demand) has trends, cycles, and dependence on external busniess variables too. This situation is tackled by another class of methods, which is called *hybrid* forecasting methods.

Following the latest stream of research, applications of machine learning to develop data driven solutions to the problems of production and operations management (Kuo and Kusiak 2019; Shen, Choi, and Minner 2019; Huang, Potter, and Eyers 2019), this study proposes a new forecasting method to address the complex demand forecasting scenarios. The proposed method is based on a state-of-the-art sequential deep learning method – long-short-term-memory networks (hereafter, LSTM) and a machine learning method – random forest (hereafter, RF). We benchmark the forecasts from the proposed method against other popular forecasting methods, and a set of relative error is employed for the sake of the empirical comparisons. The selected performance metrics evaluate the new proposition for three important characteristics of the generated forecasts: bias, accuracy, and variance. Furthermore, we conduct statistical significance tests to show the robustness of our analysis.

Our empirical data involve a multi-channel retail environment. The selected retailer owns an online store and several offline stores and sells a portfolio of different packaged food. The products offered through different channels illustrate different demand patterns and thus provide a challenging forecasting problem. The inventory replenishment cycle time for products typically ranges from one-week to one-month; therefore, the retailer requires forecasts on a weekly and monthly basis.

Figure 1 presents a broad classification of the forecasting methods based on two important dimensions viz. volume and dimensionality of the data. The multivariate data of daily unit sales for online stores and weekly sales for sixteen offline stores are available for 2.5 years. Therefore, we have a large amount of data, but this cannot be called big data. So, we can position the problem around the center of Figure 1. The methods mentioned in the quadrants of Figure 1 are self-explanatory except Quadrant 1, *towards big data*. Quadrant 1 indicates that high volume and high-dimensional data requires a big data architecture as well as algorithms and methods specially designed for a distributed big data environment. The methods available for our type of data are limited because this type of data requires

modeling both temporal as well as explanatory variables. Therefore, the forecasting problem we are facing requires a hybrid approach involving multiple types of methods, as we illustrate in Figure 1.

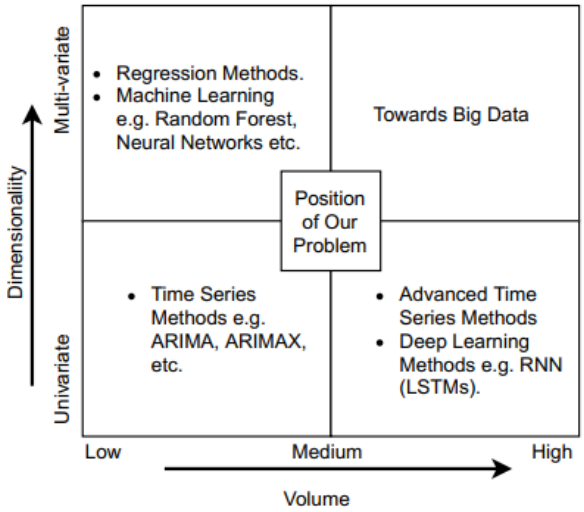


Figure 1: Classifications of Forecasting Methods by Data Characteristics

Our paper contributes to the literature in the following ways. We developed a novel hybrid method to tackle the problem when time-series data with some additional variables are available. It helps to model both types of information available in the data, whether it is in the temporal form or the form of correlation between demand and explanatory variables. Our research provides a detailed application of deep learning techniques, i.e., LSTM networks to model the time-series data. The architecture, data format and hyperparameters optimization for successful implementation of LSTMs networks are discussed in detail. Further, the application of random forest after the LSTM network is a unique feature of our method which not only improves the accuracy but also helps in understanding important variables for the behavior of demand. Also, the application of the proposed method in online and offline retail formats is provided for multi-channel retailing.

The remainder of the paper is organized as follows: Section 2 revisits and critically synthesizes the respective body of literature; Section 3 details the proposed forecasting method, while Section 4 presents the data, the tools and the measures of forecasting accuracy used. In Section 5, we present the analysis, results, and discussion of our study. Finally, Section 6 provides the main conclusions and outlines avenues for future research.

2. Literature Review

2.1 The forecasting methods

The selection of the right forecasting method is considered the ‘holy grail’ of forecasting. Along with the accuracy of the forecast the bias, variance, as well as interpretability of the forecasting model and results, are also important factors for the selection of a particular forecasting method (Nikolopoulos et

al. 2016). In this literature review, we will discuss the forecasting methods and their application as well as limitations which will help to understand and select the right type of forecasting method for a specific type of data. Furthermore, we discuss the application of forecasting methods in retail, more specifically, multi-channel retail.

One forecasting method that is quite popular among academics (as benchmark) and practitioners is Auto-Regressive Integrated Moving Average (ARIMA), with many applications in supply chain (Gilbert 2005; Svetunkov and Boylan 2019). ARIMAX is the natural extension of ARIMA when explanatory variables are available in business (Dellino et al. 2018). In forecasting problems, relationships can be linear or non-linear (Sugihara and May 1990; Terui and Van Dijk 2002). For linear relationships, ARIMAX variations perform adequately (Pai and Lin 2005) but non-linear relationships are far more challenging. In most of the cases, non-linear models are capable of capturing only specific types of non-linearity and fail to provide generic models (Wu 2010). Artificial neural networks established themselves as one of the most popular forecasting methods exactly because they can capture the various non-linearities in the data (Khashei and Bijari 2011).

Recently, other machine learning methods gained popularity which consider the forecasting problem as a regression problem, and model patterns in the target variable (dependent variable) based on correlations with the predictors – the independent variables. However, machine learning methods find it challenging to model the temporal patterns (trend and cyclicity) of time-series data, and therefore statistical time-series methods have the edge (Assimakopoulos and Nikolopoulos 2000). There is a need for new methods that can model the temporal patterns (trend and cyclicity) as well as take the benefits of the additional available features to improve accuracy of model.

Building on the success of machine learning methods, particularly neural networks, several hybrid methods that can take advantage of both time-series analysis and explanatory variables were proposed in the literature (Guo et al. 2011; Taskaya-Temizel and Casey 2005). Several attempts have been made to integrate ARIMA along with machine learning methods (Taskaya-Temizel & Casey, 2005; Khashei & Bijari, 2011). The logic behind the success of hybrid methods is that suitable methods can be employed to forecast each of the components separately, and forecasts from both components can be combined to provide a final forecast; in many instances, the forecasting performance is very promising on the real-world data (Azevedo and Campos 2016; Guo et al. 2011; Wu 2010).

Recently deep learning neural networks have shown promising results in non-linear sequence learning problems. Deep learning is a new area of research in machine learning which uses deep neural networks to accomplish the task of developing artificial intelligent models/machines. Notably, RNNs and LSTM networks are one of the most popular deep learning techniques and outperformed popular machine learning methods for time-series forecasting (Lv et al. 2015; Fischer and Krauss 2018). RNN and LSTM networks, contrary to other neural networks, have the property of retaining the information across time

steps (Hochreiter and Schmidhuber 1997). Furthermore, improved versions of LSTM networks such as a full gradient version by Graves and Schmidhuber (2005) made LSTM networks a suitable choice for non-linear sequence forecasting because it overcomes the problem of *vanishing gradient*. This updating of LSTM networks made it possible to retain information across long time steps which enable LSTM networks to use for sequence learning LSTM (Graves and Schmidhuber 2005).

Fischer and Krauss (2018) presented the application of deep learning for prediction of returns on investments in the stock market and the application of LSTM has shown significant improvements in the forecasting results. Several other works in the area of deep learning methods highlight the potential of these methods in forecasting tasks and ask for more research to be thrown towards that direction. Answering this call and to some extent corroborating to this stream of research as well extending it, we propose the development of a new such method, however, in a hybrid form to bear the advantages of more than one family of methods.

2.2 The forecasting in retail

Retailer, irrespective of online and offline retails, requires demand forecasts to support sales management (Thomassey 2010), capacity management (Aviv 2007; Doganis, Aggelogiannaki, and Sarimveis 2008), assortment planning (Dzyabura and Jagabathula 2018), order picking (Gils et al. 2017) and for several other important decisions. The demand forecasts also have significant impact on inventory ordering policies in production and retail (Doganis, Aggelogiannaki, and Sarimveis 2008), and several models are presented in the literature which highlight the importance of demand forecast for the inventory management (Erlebacher 2000; Priore et al. 2019). Moreover, demand forecasts also helps in planning the distribution, routing and logistics management in retail (Sillanpää and Liesiö 2018; Winkelhaus and Grosse 2020; Liu et al. 2020). Such importance of demand forecasts requires the retailer to achieve maximum accuracy in forecasting as that will lead to less uncertainty and better decisions.

Application of traditional time-series methods such as exponential smoothing, ARIMA, was quite popular for demand forecasting in the offline retail (Basallo-Triana, Rodríguez-Sarasty, and Benitez-Restrepo 2017). Recently, data-driven approaches like neural networks (Alon, Qi, and Sadowski 2001) and random forests for multivariate data were also explored by the researchers (Ferreira, Lee, and Simchi-Levi 2016). The research on multi-channel especially online retail is in the emerging phase (Hübner, Holzapfel, & Wollenburg, 2016). Specifically, limited literature is available for demand forecasting in multi-channel retailing. The available studies focus on analyzing the impact of demand functions viz. deterministic or stochastic (Cao, So, and Yin 2016) rather than tackling the challenge of data-driven demand forecasting. The challenge in multi-channel online-and-offline retailing is to decide upon the right method to forecast demand in the presence of multiple streams of demands. This

challenge of developing a demand forecasting model for different demand patterns is solved in this paper through a hybrid method.

3. A new hybrid deep learning forecasting method

A demand series, X_t, can be considered to have two types of variations, time-dependent, T_t, and external variable dependent, I_t. First, LSTM network is applied to forecast the demand series by capturing temporal component, T_t. After that, the residuals from the LSTM were calculated. The residuals contain the information which could not be captured by the LSTM network. A machine learning model between errors as the dependent-variable and external variables as independent variables, I_t, is used to cover this gap. The proposed method derives forecasts via a process including three stages as follows: 1) modeling linear and non-linear temporal relationships using an LSTM network, 2) modeling the non-temporal relationships as a supervised learning problem, and 3) deriving a final forecast via aggregating the forecasts generated in (1) and (2). The hybrid method is expected to be superior to multivariate LSTMs too because it contains a separate multivariate method. This separate method will avoid the limited time-window based input data and will have an undivided training dataset to model the relationships among demand and independent variables. These three parts are discussed in detail in the following subsections.

3.1 The LSTM network

The selection of the LSTM network for forecasting in retail is based on several reasons. Some of them are as follows. LSTM networks recently have shown promising results in time-series forecasting tasks (Fischer and Krauss 2018). LSTM networks are capable of working well on linear and non-linear time-series (Chollet, 2017). Therefore, decomposition of time-series into linear and non-linear components is not required. Our data contains several demand patterns generated from online and offline sales. Thus, we conjecture that LSTM networks will handle the linear/non-linear demand variations well which will eliminate the need for different methods for different demand series.

The architecture of an LSTM network memory cell

LSTM networks belong to the class of recurrent neural networks (RNNs). RNNs have the property of information persistence - i.e., retaining the state variables across time steps (Graves and Schmidhuber 2005), thus making sequential learning over time steps feasible. The architecture of an RNN is presented in Figure 2, where X_t is the input, S_t is the hidden state of the cell and H_t is the output of the RNN cell at time t. RNN can only handle short-term dependencies because it suffers from vanishing gradient problem. The LSTM networks, on the other hand, have the capability to learn long-term dependencies (Hochreiter and Schmidhuber, 1997).

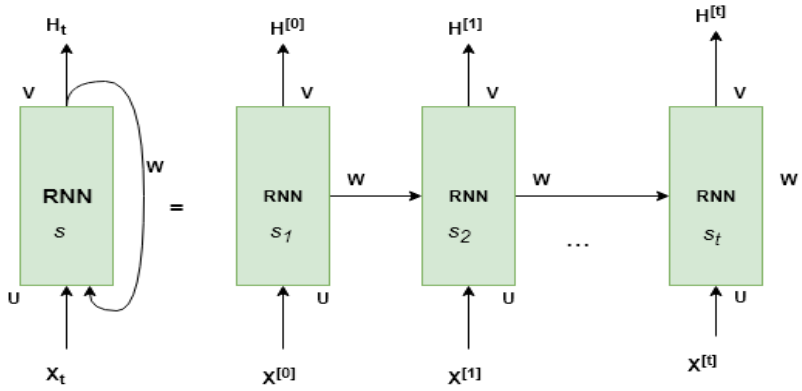


Figure 2: Recurrent Neural Network

The architecture of the LSTM network has three types of layers: 1) an input layer with a number of neurons equal to the number of input variables, 2) single or multiple hidden layers and 3) an output layer with a number of neurons equal to the number of output variables. The hidden layers of LSTM networks consist of a memory cell. LSTM networks are superior to standard RNN due to the presence of this memory cell, which helps to retain information across time steps as this was not possible in earlier neural networks. The structure of the memory cell has three types of gates[2]: 1) a forget gate (f_t), 2) an input gate (i_t), and 3) an output gate (o_t). We can see the complete architecture of the memory cell in Figure 3. In memory cell, at each time step *t*, the input consists of an element from the input sequence (X_t) and the output of the previous time step (h_{t-1}). At cell state *t*: a) the forget gate takes these inputs and decide upon which information will be removed from memory, b) the input gate decides which information shall be added to memory (at cell state *t*), and c) the output gate decides the output of the memory block.

LSTM network processes information through a sequence of four steps:

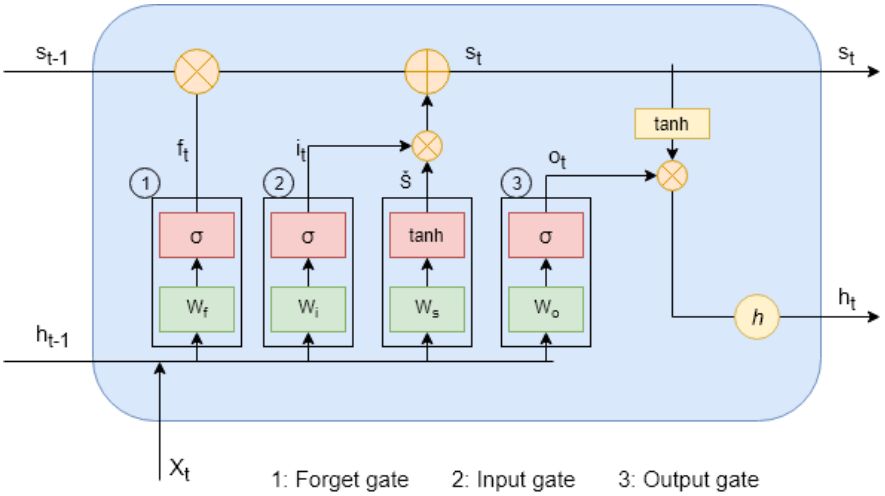
In the first step, a sigmoid (a non-linear activation function) layer called forget gate layer, which takes X_t and h_{t-1} as inputs and b_f as bias, computes the vector of activation values, f_t, for each of the

[2] Notation used in this section:

- X_t: the input vector at time step *t*
- h_t : the output vector at time step *t*
- s_t : the vector for cell state *t*
- $\tilde{s}_t$: the vector for a candidate value for input gate
- $b_f, b_i, b_{\tilde{s}}, b_o$: bias vectors
- $\sigma(.)$: denotes the sigmoid function $\left(f(x) = \frac{1}{1+e^{-x}}\right)$
- $W_{f,x}, W_{f,h}, W_{\tilde{s},x}, W_{\tilde{s},h}, W_{i,x}, W_{i,h}, W_{o,x}, W_{o,h}$: weight matrices for input and outputs for the three gates
- f_t, i_t, o_t : vectors of values obtained after activation of the gates (forget gate, input gate and output gate)

values in cell state s_{t-1} within a normalized range between 0 (completely get rid-off) to 1 (completely keep). Then the activation value vector is calculated as follows:

$$f_t = \sigma\left(W_{f,x}X_t + W_{f,h}h_{t-1} + b_f\right) \quad (1)$$



	Product 1								
Week	1	2	3	...	101	102	103	104	...
Units	27	36	28	...	48	44	30	27	...

	Product 1					
Seq. 1	1	2	3	...	101	102
	27	36	28	...	48	44

	Product 1					
Seq. 2	2	3	...	101	102	103
	36	28	...	48	44	30

Figure 3: The LSTM Memory Cell Architecture and the input data to the LSTM

In the second step, it is decided which information will be added to the memory cell state s_t. This step has two parts: first, candidate values $\tilde{s}_t$ are calculated. In the second step, an activation layer called the *input gate layer*, calculated as follows:

$$\tilde{s}_t = tanh\left(W_{\tilde{s},x}X_t + W_{\tilde{s},h}h_{t-1} + b_{\tilde{s}_t}\right) \quad (2)$$

$$i_t = \sigma\left(W_{i,x}X_t + W_{i,h}h_{t-1} + b_i\right) \quad (3)$$

In the third step, we update the cell state using new information. We use the Hadamard product in this step:

$$s_t = f_t.s_{t-1} + i_t.\tilde{s}_t \quad (4)$$

In the last step, h_t, we calculate the output of the memory cell as follows:

$$o_t = \sigma\left(W_{o,x}X_t + W_{o,h}h_{t-1} + b_o\right) \quad (5)$$

$$h_t = o_t * \tanh(s_t) \quad (6)$$

As shown in Figure 3, the input variables (demand and lagged variables) are inserted to the LSTM's input gate. LSTM network process the inputs step by step using Equations (1)-(6), and after completion of the process, it generates the final output sequence. The output vector is then the forecast of the network. LSTM networks are trained in multiple iterations known as epochs. During these iterations, bias and weights change to minimize the objective function across the training sets. For our task, the mean absolute error (MAE) is used as loss functions and advanced hyper-parameter optimization through grid search is used to decide the final parameters of the prediction model (Bergstra, Yamins, and Cox 2013). Further details of hyper-parameter optimization are explained in Section 5.1 during empirical analysis.

3.2 The covariates model

Random forests (RF) are selected because, in retail, non-temporal information represents non-cyclical and irregular variations in demand data due to the impact of promotions and other markdown events or holidays. We could employ multiple regression for the aforementioned task, but we decided to use RF due to their superiority in accuracy with wide applicability in retail management (Ferreira, Lee, and Simchi-Levi 2016). Moreover, RF presents very competitive forecasting performance in recent empirical evaluations (Ferreira, Lee, and Simchi-Levi 2016; Pierola, Epifanio, and Alemany 2016).

RF was initially proposed by Ho (1995) and later fully developed by Breiman (2001). RF uses the bagging strategy (i.e., bootstrap aggregation), which is to generate random samples of data and train different decision trees on these samples. The predictions from these different trees are then aggregated to generate the final predictions. During training, RF is allowed to select a random subset of variables to split each node and to grow trees to more depth with smaller leaf size. These strategies help RF to overcome the challenges of decision trees (e.g., overfitting) and provides a robust method for prediction. For applying RF in the proposed forecasting method, explanatory variables are used as input variables, and residuals from stage 1 are used as the dependent variable. The RF model is trained over these data and generates forecasts for residuals, which will be used later at the final stage.

3.3 *The Forecast*

The final step of the proposed method is the aggregation of the forecasts from stages (1) and (2). The forecasts are aggregated by taking the sum of both forecasts; this is an aggregation and not a combination as in stage 2 we model the residuals of stage 1. Thus:

1) The univariate demand series is converted to sequences of inputs for LSTM network, and forecasts ($\hat{X}_t^1$) are generated from it.

2) The residuals from the first forecast are calculated as $r_t = X_t - \hat{X}_t^1$ and regressed over independent variables via the RF model and forecasts ($\hat{X}_{t,r}^2$) for the residuals are obtained.
3) The final forecast is obtained as: $\hat{X}_t = \hat{X}_t^1 + \hat{X}_{t,r}^2$.

The proposed hybrid method is expected to perform better than its constituents, because it overcomes the limitations of the both LSTM and RF. LSTM typically iterate over a set of past values to predict the future values, known as *batch size* in the LSTM. The batch size can vary from one sample to whole training data but smaller batch sizes (e.g. 32, 48, 64, etc.) are used for best results. Therefore, LSTM can not take advantage of data available across the products to train the model. On the other hand, RF can be trained over data on all products and thus, have more data and able to model the relationships demand and independent variables; but RF can not model the trend and cyclicity in demand data. In this context, out proposed method with above mentioned three step-approach overcomes these limitations and provide a complete solution to model temporal and regression effects of the demand data. Moreover, the errors are expected to minimized as the proposed method combines the outputs from two best methods (Pierola, Epifanio, and Alemany 2016). Further, we restricted the model to point forecasts, however, using bootstrap sampling the prediction interval can be generated (Antipov and Pokryshevskaya 2012).

4. The Experimental Setup

4.1. The Data

The dataset is from a multi-channel retailer selling packaged food products through one online platform and eleven offline stores spread across a metro city and covers a period of 30 months. The daily sales data is available for the online store, and weekly sales data is available for offline stores. The dataset consists of attributes related to sales (base price, sales, etc), products (size, brand, volume, etc), and stores (visits, sales area, etc). We also have promotional activity information. These variables are listed in Table 1 (B). We also calculate the "relative price" of the products. The relative price of the product is the ratio of "price of a product" and "average price of similar products from other brands". The relative price for an item $S(s,i)$ in a sub-segment *s,* which has N number of items from N different brands is given by:

$$relative\ price\ of\ an\ item, S(s,i) = \frac{price\ of\ S(s,i)}{\sum_{i \in S(s)} price\big(S(s,i)\big)/N} \qquad (7)$$

The relative price is used to consider the impact of the competition within the category of products. As mentioned by Mazumdar, Raj, and Sinha (2005), the relative price tends to be a strong predictor for retail products. Ferreira, Lee, & Simchi-Levi (2016) also came to a similar conclusion for the importance of relative price in forecasting.

Table 1: (A) List of variables in the dataset, and (B) Summary statistics of variables for each year

(A)	*Products' details*	*Transactional details*	*Store's details (offline only)*
	• Category • Sub-category • Base Price • Brand • Size • Volume	• Sales • % Discount • Holiday (binary)	• Display • Visits • Feature • Store Area • Temporary price reduction (TPR) • No. of households (HHS) • Parking Available

(B)	Descriptive Statistics		
Variable Name	Median	Mean (SD)	Remark
Sales	424	452.06 (124.1)	Product Sales
Base Price	3.76	4.16 (1.89)	Selling Price
% Discount	0.00	0.23 (0.38)	Promotional Discount
Visits*	17	19.02 (23.28)	No. of customer visits
HHS*	9	14.62 (21.33)	No. of households
Sales Area*	48216	48924 (13471)	Sales area in a store
		% of cases	Remark
Feature		73.26	In in-store circular
Display		12.78	In-store Display
TPR		24.01	Temporary Price Reduction

After performing initial data cleaning, we get data for 16 product types, each available for one online platform and 11 offline stores, which leads to a total of 192 series. The descriptive statistics of the data, which can be presented in numbers, are shown in Table 1 (B). As the retailer requires demand forecasts for operational decisions with a planning horizon of one day, one week or one month, therefore, we keep the last one-month as a test dataset for out-of-sample daily, weekly, and monthly evaluations, and utilized the rest of the data for training and validation.

4.2 Hardware and Software

We perform the data management and analysis entirely in two freeware and open-source platforms: R and Python. We use deep learning library Keras (Chollet, 2015), which runs on top of TensorFlow, CNTK, and Theano for implementing LSTM networks in Python. We use the python libraries hyperopt (Bergstra, Yamins, and Cox 2013) and hyperas with Keras for the implementation of the grid search algorithm to find optimal settings of hyperparameters in LSTM networks.

We use the popular *randomforest* package (Liaw and Wiener 2002) in R for fitting the RF model. For the benchmarking models, we use the *forecast* package (Hyndman et al. 2015) for ARIMAX via the auto.arima( ) function and xreg parameter; furthermore, we use the *neuratnet* (Günther and Fritsch

2010) package for neural networks, and the *caret* (Kuhn 2008) package for training and evaluating the models. All the modeling and analysis are performed in a core-i7 processor with 8 GBs of DDR4 RAM, and 4GB of GPUs. Notably, LSTM networks are trained on GPUs.

4.3 The metrics

We do empirically evaluate the forecasting performance of our proposed method for:

- *Bias* – to check the forecasting method for its tendency of over-forecasting or under-forecasting of actual values.
- *Accuracy* – to check how closely forecasted values are to the actual values.
- *Variance* – to check the average deviation of the forecast from the mean forecast.

The metrics used to measure the characteristics mentioned above are as follows: mean error (ME) for bias, mean absolute error (MAE) for accuracy, and mean squared error (MSE) for variance (Nikolopoulos et al. 2016). To avoid any scaling issues, we do employ the relative versions of these aforementioned metrics, thus the relative mean error (RME), relative mean absolute error (RMAE) and the relative mean squared errors (RMSE). We calculate the relative errors by dividing the sum or mean of errors from the evaluated method with that of a benchmark method - the naïve method in this instance. To test the statistical significance of our empirical results, two tests are used: a) the Pesaran and Timmermann (1992) (PT) test, and b) the Diebold and Mariano (1995) (DM) test is used.

We calculate the relative over multiple time-series, and therefore average relative forecast errors are used. Overall three metrics and two statistical tests are used to benchmark the performance of all forecasting methods and these are: average relative mean error (ARME), average relative mean absolute error (ARMAE), average relative mean squared error (ARMSE), and the Pesaran and Timmermann (1992) test and the Diebold and Mariano (1995) test.

5. Empirical Results

In this section, we present, compare and discuss the forecasting performance of our model. We produce forecasts for one-week and four weeks ahead, and we benchmark our method against random forests (RF), neural network (NN), ARIMAX, multiple linear regression (MLR), and the LSTM network.

5.1 Implementation

To implement LSTM networks using Keras, the input data is transformed into a specific 3D shaped vector which takes the following form: [samples, timesteps, features]. Also, we do normalize the variables in the range of [-1,1] as Keras works best with variables in this range. For the training of the LSTM network, we use *hyperparameter optimization*. In this grid search, the following alternatives are used for creating the search space for the selection of hyperparameters:

- For optimizers: “rmsprop”, ‘adam”, and “sgd”
- For the activation function: “linear”, “relu” and “tanh” functions.
- A set of values to find the optimum number of LSTM layers for the network.

The optimizers, root mean square propagation (rmsprop) (Hinton, Srivastava, and Swersky 2012), adaptive moment estimation based adam method (Kingma and Ba 2017), and stochastic gradient descent (sgd) , and activation function, linear $(f(x) = x)$, rectified linear unit (relu) $\left(f(x) = \begin{cases} 0 \; for \; x \leq 0, \\ x \; for \; x \geq 0, \end{cases}\right)$, and tanh $\left(f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}\right)$, are used as alternatives because these are widely recommended in literature for forecastthe ing purposes (Fischer and Krauss 2018; Lv et al. 2015). For hyperparameter optimization, we then use the Bayesian optimization algorithm - a Tree-structured Parzen Estimator Approach (TPE) developed by Bergstra et al. (2011). For the implementation of this algorithm, we use the python libraries hyperopt and hyperas with Keras (Bergstra, Yamins, and Cox 2013).

Furthermore, regularization and early stopping techniques are used for training to prevent the overfitting. In regularization, we drop some input units at input gates and at the recurrent connection in the LSTM network. Thus, it helps to create models that can generalize better (Gal and Ghahramani 2016). We also apply early stopping techniques, where we decide to stop the training of the network if the improvement over the incremental training step becomes negligible. For regularization, rather than having one random value, we used a uniform distributed value of dropouts (between 0 and 1), and for early stopping, we used a big patience value, i.e. the allowable number of steps with negligible improvement before stopping the algorithm. After the training of LSTM networks through grid search, the best LSTM network model based on validation accuracy is selected to use it for the out-of-sample prediction on the test data. Using these experimental settings, the optimal settings for hyperparameters are obtained, and after manually checking the training and validation accuracy curves, the final configurations are selected.

For the RF part of the method, we apply repeated cross-validations in order to avoid overfitting and at the same time obtain best-fit models. Similarly, for benchmark methods, the best possible configuration is used. The mean forecast is used as the naïve method for finding the relative errors for all the methods.

5.2 The demand forecasting module for the multi-channel retailer

In multi-channel retail, a product is offered through multiple channels and the often factors like price and discount for the product vary on these channels and so do the demand for the product. Therefore, retailer requires different forecasting models for the same product for different channels. Another challenge in multi-channel retail is that it has several purchasing channels and fulfillment channels and the key question is how to segregate the demand for proper demand planning for channels. One

approach is to forecast along each demand channel. However, this makes demand forecasting a cumbersome task yet it will generate very irregular demand patterns, especially on some less popular channels, that will be very hard to learn for any algorithm and will result in very low-accuracy forecasts. Also, the benefits of aggregate demand will forego too. Therefore, the solution is to forecast demand for a product based on its order origination, i.e. online or offline. This method will streamline the process of demand forecasting and other decision making like pricing in multi-channel. The architecture for the demand forecasting module is provided in Figure 5. The present architecture will serve as a decision support tool for the demand forecasting and management in multi-channel retail.

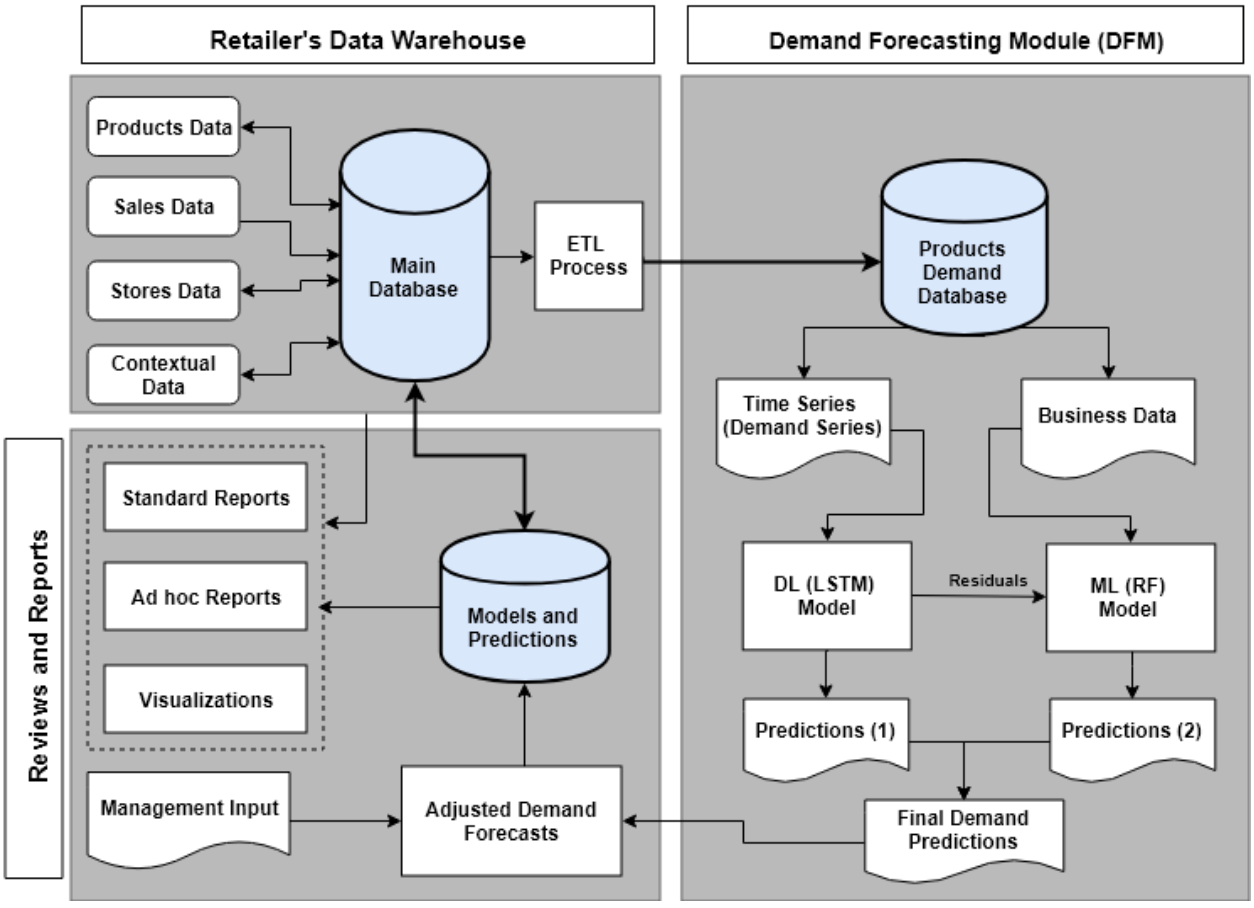


Figure 4: Demand Forecasting Tool for Multi-channel Retailer

5.3 Results for the online channel: The Online Store

The proposed forecasting method is first applied to the data from the online store. The data consists of sixteen demand series with independent variables. The average one-week ahead forecasting errors from each forecasting method used in the study are presented in Table 2. These empirical results show that the forecasts from the proposed method are more accurate; less biased and have less variance. These results for short-term forecasting horizons show that the proposed method is outperforming the benchmark methods on all three performance characteristics (with the respective metrics been: ARME, ARMAE, ARMSE).

	RF	NN	ARIMAX	MLR	LSTM	**Proposed**
ARME	2.6682	1.7738	1.1661	3.5401	0.2007	*0.0762*
ARMAE	0.9455	1.6302	0.7671	1.0420	0.5268	*0.5003*
ARMSE	0.7722	1.6879	0.5384	0.9684	0.2434	*0.2167*

Table 2: Online channel; one-week ahead: ARME, ARMAE, and ARMSE
(the smaller the better; with *italics* the top performer)

Table 3 presents the average relative forecasting errors for the sixteen products for the mid-term forecasting horizon of 4 weeks (approximately one-month ahead). The results show that the proposed method is yet again the most accurate followed by LSTM. The empirical results are the same across all three forecasting performance metrics. The actual extrapolations from our method and four benchmarks for one of the sixteen products are illustrated in Figure 5. It can be learned from Figure 5 that time-series methods (or sequence modeling methods) such as ARIMAX, LSTM and the proposed methods are better performing in forecasting the pattern than the machine learning methods. The machine learning method is able to forecast some peak demand which may be due to information available in explanatory variables available in data.

	RF	NN	ARIMAX	MLR	LSTM	**Proposed**
ARME	0.8162	0.8682	0.3612	0.7640	0.3356	*0.1787*
ARMAE	0.9931	1.7073	0.7797	1.0883	0.6267	*0.5913*
ARMSE	0.9748	1.9593	0.5849	1.0938	0.4031	*0.3755*

Table 3: Online channel; one-month (4 weeks) ahead: ARME, ARMAE, and ARMSE
(the smaller, the better with *italics* the top performer)

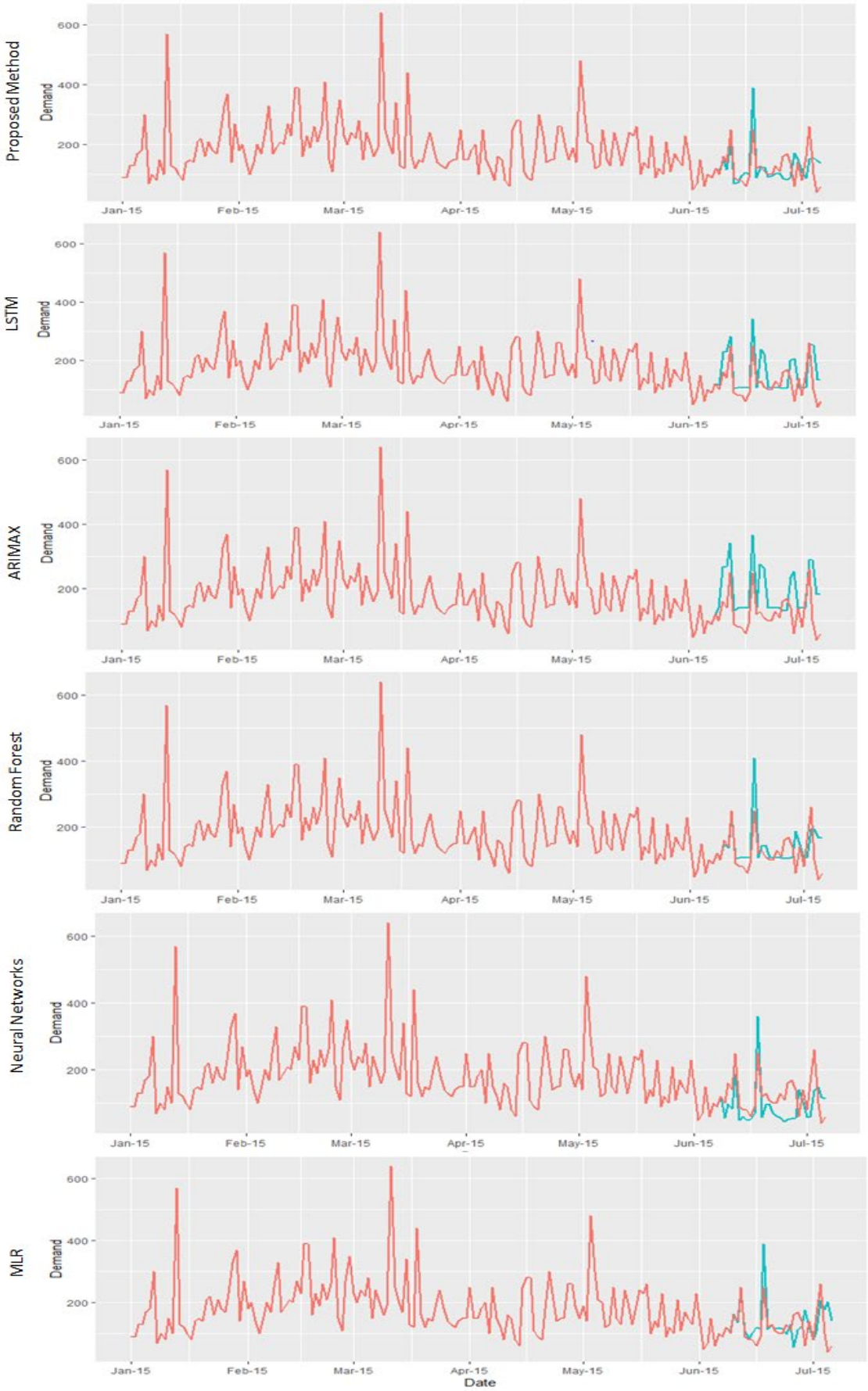


Figure 5: Extrapolations from the Proposed Method and Benchmark Methods

Moreover, we do perform statistical significance tests for the empirically estimated difference between the accuracy (forecasting errors) of our method and the aforementioned benchmarks with the DM and PT tests (table 4). We can not aggregate p-values across different products. Thus, we present results for weekly predictions for Product 1 in Table 4. Our method always outperforms all the other methods and the differences in the forecasting performance are statistically significant at a 95% confidence level[3]. We do get similar results for the entire set of products in our analysis.

A: DM Test							**B: PT Test**	
i *j* =	**Proposed**	RF	NN	ARIMAX	MLR	LSTM	Method	Result
Proposed	-	0.0000	0.0000	0.0026	0.0000	0.0061	**Proposed**	0.0000
RF		-	0.0114	1.0000	0.0000	1.0000	RF	0.0000
NN			-	1.0000	0.9984	1.0000	NN	0.0000
ARIMAX				-	0.0000	1.0000	ARIMAX	0.0000
MLR					-	1.0000	MLR	0.0000
LSTM						-	LSTM	0.0000

Table 4: Online channel; Statistical significance tests for Product #1.
Panel A: DM test; Panel B: PT test.

Panel A in table 4, the DM test, shows the p-values for the null hypothesis that the paired methods have equal performance. All individual hypothesis is rejected at 95% significance level over, and therefore, the forecasts from the proposed method are superior to all benchmark methods: RF, NN, ARIMAX, MLR, and LSTM. In panel A of Table 4, the p-values illustrate our confidence that method *i* have inferior prediction accuracy than method j. In panel B, we see the p-values for the PT test for the null hypothesis that actual data and forecast values are independently distributed. So, we can assume with high confidence that prediction and response from the forecasting method are independently distributed.

5.4 Results for the offline channel: The Physical Stores

After applying the proposed method to the online retail data, we focus our analysis on the offline stores. For offline stores, demand data are available on a weekly basis. The number of variables is higher in this case as the retailer holds more information regarding stores and offline customer orders. In addition, the additional store related information like visits, display, etc. (Table 1) enables us to consider the impact of local factors at the store level.

The data for sixteen products from eleven different offline stores is available. In total, we have one hundred and ninety-two (192) demand series (16 for the online channel and 176 for the offline channel), and for each of these series, we have collected the respective independent variables. The average relative

[3] We also get similar results if we test only the lower 50% of the worst performing forecasts for our method and the results are always statistically significant at 95% level of confidence.

forecasting errors in the offline channel, from the application of each of the six forecasting methods, for one-week and one-month ahead are presented in Tables 5 and 6 respectively:

	RF	NN	ARIMAX	MLR	LSTM	**Proposed**
ARME	0.3058	0.4706	0.4468	0.6375	0.4931	*0.2133*
ARMAE	0.3992	0.6171	0.7703	0.7582	0.6020	*0.3616*
ARMSE	0.2739	0.5804	0.6582	0.6809	0.5063	*0.2284*

Table 5: Offline channel; one-week ahead: ARME, ARMAE, and ARMSE
(the smaller the better with *italics* the top performer)

	RF	NN	ARIMAX	MLR	LSTM	**Proposed**
ARME	0.2873	0.4375	0.5532	0.8415	0.5757	*0.2397*
ARMAE	0.4031	0.5248	0.6621	0.7473	0.6190	*0.3579*
ARMSE	0.2510	0.3201	0.5802	0.6983	0.5102	*0.2165*

Table 6: Offline channel: one-month (4 weeks) ahead: ARME, ARMAE, and ARMSE
(the smaller, the better with *italics* the top performer)

For the short-term horizon of one-week ahead, we yet again see that on average[4] across the eleven physical stores, our proposed is the top-performing one across all three metrics. RF is the second best and LSTM the third one except for ARME where ARIMAX is third. For the mid-term horizon of four weeks ahead, we yet again see that on average across the eleven physical stores, our proposed method is the top-performing one across three metrics: for ARME, ARMAE and ARMSE presenting the more accurate forecasts with the least bias and variance. RF is the second best and LSTM the third one apart from ARME where ARIMAX is third.

A: DM Test							**B: PT Test**	
i *j* =	**Proposed**	RF	NN	ARIMAX	MLR	LSTM	Method	Result
Proposed	-	0.0986	0.0000	0.0000	0.0000	0.0000	**Proposed**	0.0000
RF		-	0.0000	0.0000	0.0000	0.0000	RF	0.0000
NN			-	0.0000	0.0000	0.1592	NN	0.0000
ARIMAX				-	0.2961	1.0000	ARIMAX	0.0000
MLR					-	1.0000	MLR	0.0000
LSTM						-	LSTM	0.0000

Table 7: Offline channel; Statistical significance tests for Product #1 at Store #1.
Panel A: DM test; Panel B: PT test.

Further, the DM and PT tests are conducted to test the statistical significance of the results for offline data also. Table 7 present the results for the weekly prediction for Product 1 at store 1. Similar results

[4] We have seen no differences in the forecasting performance across the eleven stores; thus, we report the average

are obtained for all the products across all stores. The proposed method outperformed the other methods at a 95% confidence level. An important aberration from online results is that the MLR is performing better than other methods and in some cases close to RF and NN. It can be attributed to additional exogenous variables available in offline stores. This also confirms the superiority of our method for demand prediction in offline retail also.

5.5 Discussions

The proposed forecasting method and benchmark forecasting methods were applied to a total of 16 online demand series and 176 offline demand series. Holistically, based on three metrics and two statistical tests, it has been proved that the proposed forecasting method is outperforming other benchmarking methods on all characteristics. For the online channel, where limited independent variables are available, LSTM is the second-best performing method. ARIMAX has the worst performance of all methods, and this can be attributed to the inability of ARIMAX to model nonlinear temporal information. This implies that online demand patterns are more complex to model by linear time-series methods such as ARIMAX. Whereas, advanced methods such as the proposed method and LSTM are well suited for the demand forecasting in the online environment. For the offline channel, where sufficient independent variables were available, RF came second to the proposed method. The weekly aggregation of demand and the presence of additional features are helping the random forest to perform better than other methods. This is also in agreement with the present literature which recommends the random forest for the demand forecasting in offline retail (Qu et al. 2017). However, the proposed method is performing better than even a random forest. This is so because the hybrid method is a combination of both LSTM and random forest which enables it to benefit it from the complementary strengths of the two methods.

The random forest part of the proposed method provided the much-needed edge in forecasting accuracy by predicting accurately the sudden changes (say, due to holidays, promotions, etc.) in the forecast by using the respective independent variables. Another significant result of the random forest is the variable importance. By aggregating the variable importance results from models, the top variables are Price (100%), Relative Price (65.47%), Display (55.04%), Discount (35.27%), and TPR (9.51%). Similarly, the relative importance is calculated from the RF method and it is found that top variables are Price (100%), Relative Price (75.43%), Display (38.34%), Discount (14.90%) and TPR (8.718%). We observed that while the order of relative importance is the same from the proposed method and RF, but the relative importance is different. It is so as LSTM filtered out the temporal effects like the start of the week, month or weekend, from the sales data and thus, the results from the proposed method are only of the relationship between independent variables and sales. That is why the Relative price, Display, and Discount have more relative importance than in the case of RF, where importance is highly skewed towards Prices. These are quite helpful results for a multi-channel retailer as these can help not

only in managing promotional events or sales but also in decisions like assortment planning. The enhanced variable importance is also a useful characteristic of the proposed method.

6. Conclusions, managerial insights, and further research

In this paper, a new - hybrid in nature - forecasting method is proposed based on an LSTM and an RF. To benchmark, the performance of the proposed method, a set of popular and widely used competitive methods are tested including Naïve, MLR, NN, ARIMAX and the two components of our approach independently: RF and LSTM. Three forecasting error metrics are employed ARME, ARMAE, and ARMSE as proxies for the bias, accuracy, and variance of the evaluated forecasts. Furthermore, the DM and PT statistical significance tests are used to attest to our empirical findings.

We perform empirical evaluations over 192 time-series and respective cues of information over two different channels: an online channel for 16 products and offline channel including the same 16 products sold over 11 different physical stores. All the analyses indicate that our newly proposed method outperforms all benchmarks used in this study. The extensive empirical evidence presented here advocates the case for the potential of the use of our method in a multi-channel demand forecasting context; as well as calling for further research, testing, and development of similar hybrid methods.

The current study makes important contributions to the theory and practice of Operational Research/Management Science (ORMS) and Predictive Analytics. The first contribution is the development of a robust and flexible forecasting method, which can be used to model complex demand patterns. Second, the proposed method is hybrid in nature and first applies an LSTM network to model the temporal characteristics of the time-series and then an RF is modeling the residuals of the fitting of LSTM network to the data via employing any exogenous information in the form of cues of information available – information that differs for each respective channel. Further, the third contribution is that this study provides a new application area for innovative applications of ORMS as until recently researchers have used LSTM networks in finance and economics forecasting problems, but not in a retailing demand forecasting context. Finally, the final contribution is that this work illustrates the architecture as well as the step-by-step advanced implementation of an LSTM network for demand forecasting problems. To that end, the details of implementation and advanced hyperparameter optimization of the LSTM network are described in detail.

The obtained results have several practical implications for retail managers. The managers can use the proposed method to accurately forecasts the complex demand patterns. The use of deep learning and machine learning methods are very also efficient in managing large dataset generating from latest big data, IoT, Supply Chain 4.0, etc. business environments. The use of business data in demand planning provides an added advantage to the managers to include the significant variables according to their own judgment. Further, the relative importance of factors affecting sales of a particular category of product will help in efficiently designing the targeted promotional events, an optimal mix of assortment display,

and shelf-space optimization in the retail stores. Moreover, accurate demand forecasts will lead to better ordering policies and thus, minimization of inventory management cost, and optimal distribution and logistics planning for satisfying the future demand.

As far as future research in this domain is concerned, we believe that the following ideas are worth pursuing further. Also, there are some limitations in the current study, which can be addressed by future research. On the methodology side, the new and more advanced neural networks such as a convolutional neural network (CNN) or spiking neural networks should be adapted for time-series forecasting problems. Notably, CNN would be potentially successful in handling big data. Another interesting future research direction will be to compare the performance of LSTM networks with the deep learning neural networks. These efforts will corroborate further to the new stream of research of deep learning methods for forecasting. For further advancement in demand forecasting in a multi-channel retail context, more data on non-traditional fulfillment channels viz. order online and home delivery, order online and pick up at the pickup point, etc., should be gathered. By doing this, the proposed method should be tested on demand patterns generated by these new channels. Further, when demand and resources from multiple channels are seamlessly integrated – often referred to as omnichannel retail – it may not be the case that the same forecasting models will prevail and as such this will be an important topic for future research. For further generalization in the future the application of these deep learning techniques should be extended and tested on data from other domains such as healthcare, economics, transportation, etc.

References

Alon, Ilan, Min Qi, and Robert J. Sadowski. 2001. "Forecasting Aggregate Retail Sales: A Comparison of Artificial Neural Networks and Traditional Methods." *Journal of Retailing and Consumer Services* 8 (3): 147–56.

Antipov, Evgeny A., and Elena B. Pokryshevskaya. 2012. "Mass Appraisal of Residential Apartments: An Application of Random Forest for Valuation and a CART-Based Approach for Model Diagnostics." *Expert Systems with Applications* 39 (2): 1772–1778.

Assimakopoulos, Vassilis, and Konstantinos Nikolopoulos. 2000. "The Theta Model: A Decomposition Approach to Forecasting." *International Journal of Forecasting* 16 (4): 521–530.

Aviv, Yossi. 2007. "On the Benefits of Collaborative Forecasting Partnerships between Retailers and Manufacturers." *Management Science* 53 (5): 777–794.

Azevedo, Vitor G., and Lucila M. S. Campos. 2016. "Combination of Forecasts for the Price of Crude Oil on the Spot Market." *International Journal of Production Research* 54 (17): 5219–35. https://doi.org/10.1080/00207543.2016.1162340.

Basallo-Triana, Mario José, Jesús Andrés Rodríguez-Sarasty, and Hernán Darío Benitez-Restrepo. 2017. "Analogue-Based Demand Forecasting of Short Life-Cycle Products: A Regression Approach and a Comprehensive Assessment." *International Journal of Production Research* 55 (8): 2336–50. https://doi.org/10.1080/00207543.2016.1241443.

Bergstra, James S., Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. "Algorithms for Hyper-Parameter Optimization." In *Advances in Neural Information Processing Systems*, 2546–2554.

Bergstra, James, Dan Yamins, and David D. Cox. 2013. "Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms." In *Proceedings of the 12th Python in Science Conference*, 13–20. Citeseer.

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

Cao, J., K.C. So, and S. Yin. 2016. "Impact of an 'Online-to-Store' Channel on Demand Allocation, Pricing and Profitability." *European Journal of Operational Research* 248 (1): 234–45. https://doi.org/10.1016/j.ejor.2015.07.014.

Chollet, François. 2015. *Keras*.

Chollet, Francois. 2017. *Deep Learning with Python*. Manning Publications Co.

Dellino, Gabriella, Teresa Laudadio, Renato Mari, Nicola Mastronardi, and Carlo Meloni. 2018. "A Reliable Decision Support System for Fresh Food Supply Chain Management." *International Journal of Production Research* 56 (4): 1458–85. https://doi.org/10.1080/00207543.2017.1367106.

Diebold, Francis X., and Roberto S. Mariano. 1995. "Comparing Predictive Accuracy." *Journal of Business & Economic Statistics*, 253–63.

Doganis, Philip, Eleni Aggelogiannaki, and Haralambos Sarimveis. 2008. “A Combined Model Predictive Control and Time Series Forecasting Framework for Production-Inventory Systems.” *International Journal of Production Research* 46 (24): 6841–53. https://doi.org/10.1080/00207540701523058.

Dzyabura, D., and S. Jagabathula. 2018. “Offine Assortment Optimization in the Presence of an Online Channel.” *Management Science* 64 (6): 2767–86. https://doi.org/10.1287/mnsc.2016.2708.

Erlebacher, S.J. 2000. “Optimal and Heuristic Solutions for the Multi-Item Newsvendor Problem with a Single Capacity Constraint.” *Production and Operations Management* 9 (3): 303–18.

Ferreira, Kris Johnson, Bin Hong Alex Lee, and David Simchi-Levi. 2016. “Analytics for an Online Retailer: Demand Forecasting and Price Optimization.” *Manufacturing & Service Operations Management* 18 (1): 69–88. https://doi.org/10.1287/msom.2015.0561.

Fischer, Thomas, and Christopher Krauss. 2018. “Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions.” *European Journal of Operational Research* 270 (2): 654–69. https://doi.org/10.1016/j.ejor.2017.11.054.

Gal, Yarin, and Zoubin Ghahramani. 2016. “A Theoretically Grounded Application of Dropout in Recurrent Neural Networks.” In *Advances in Neural Information Processing Systems*, 1019–27.

Gilbert, Kenneth. 2005. “An ARIMA Supply Chain Model.” *Management Science* 51 (2): 305–10.

Gils, Teun van, Katrien Ramaekers, An Caris, and Mario Cools. 2017. “The Use of Time Series Forecasting in Zone Order Picking Systems to Predict Order Pickers’ Workload.” *International Journal of Production Research* 55 (21): 6380–93. https://doi.org/10.1080/00207543.2016.1216659.

Graves, Alex, and Jürgen Schmidhuber. 2005. “Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures.” *Neural Networks*, IJCNN 2005, 18 (5): 602–10. https://doi.org/10.1016/j.neunet.2005.06.042.

Günther, Frauke, and Stefan Fritsch. 2010. “Neuralnet: Training of Neural Networks.” *The R Journal* 2 (1): 30–38.

Guo, Zhen-hai, Jie Wu, Hai-yan Lu, and Jian-zhou Wang. 2011. “A Case Study on a Hybrid Wind Speed Forecasting Method Using BP Neural Network.” *Knowledge-Based Systems* 24 (7): 1048–1056.

Hinton, Geoffrey, Nitish Srivastava, and Kevin Swersky. 2012. “Neural Networks for Machine Learning Lecture 6a Overview of Mini-Batch Gradient Descent.” *Cited On* 14: 8.

Ho, Tin Kam. 1995. “Random Decision Forests.” In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference On*, 1:278–82. IEEE.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9 (8): 1735–80.

Huang, Shupeng, Andrew Potter, and Daniel Eyers. 2019. “Social Media in Operations and Supply Chain Management: State-of-the-Art and Research Directions.” *International Journal of Production Research*, 1–33.

Hübner, Alexander, Andreas Holzapfel, and Johannes Wollenburg. 2016. “Retail Logistics in the Transition from Multi-Channel to Omni-Channel.” *International Journal of Physical Distribution & Logistics Management* 46 (6/7): 562–83. https://doi.org/10.1108/IJPDLM-08-2015-0179.

Hyndman, Rob J., G. Athanasopoulos, S. Razbash, D. Schmidt, Z. Zhou, Y. Khan, C. Bergmeir, and E. Wang. 2015. “Forecast: Forecasting Functions for Time Series and Linear Models.” *R Package Version* 6 (6): 7.

Khashei, Mehdi, and Mehdi Bijari. 2011. “A Novel Hybridization of Artificial Neural Networks and ARIMA Models for Time Series Forecasting.” *Applied Soft Computing* 11 (2): 2664–2675.

Kingma, Diederik P., and Jimmy Ba. 2017. “Adam: A Method for Stochastic Optimization.” *ArXiv:1412.6980 [Cs]*, January. http://arxiv.org/abs/1412.6980.

Kuhn, Max. 2008. “Caret Package.” *Journal of Statistical Software* 28 (5): 1–26.

Kuo, Yong-Hong, and Andrew Kusiak. 2019. “From Data to Big Data in Production Research: The Past and Future Trends.” *International Journal of Production Research* 57 (15–16): 4828–4853.

Liaw, A., and M. Wiener. 2002. “Classification and Regression by RandomForest.” *R News* 2 (3): 18–22.

Liu, Chang, Yongfu Feng, Dongtao Lin, Liang Wu, and Min Guo. 2020. “Iot Based Laundry Services: An Application of Big Data Analytics, Intelligent Logistics Management, and Machine Learning Techniques.” *International Journal of Production Research* 0 (0): 1–19. https://doi.org/10.1080/00207543.2019.1677961.

Lv, Yisheng, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. 2015. “Traffic Flow Prediction with Big Data: A Deep Learning Approach.” *IEEE Trans. Intelligent Transportation Systems* 16 (2): 865–873.

Mazumdar, Tridib, Sevilimedu P. Raj, and Indrajit Sinha. 2005. “Reference Price Research: Review and Propositions.” *Journal of Marketing* 69 (4): 84–102.

Nikolopoulos, Konstantinos, Samantha Buxton, Marwan Khammash, and Philip Stern. 2016. “Forecasting Branded and Generic Pharmaceuticals.” *International Journal of Forecasting* 32 (2): 344–357.

Pai, Ping-Feng, and Chih-Sheng Lin. 2005. “A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting.” *Omega* 33 (6): 497–505.

Pesaran, M. Hashem, and Allan Timmermann. 1992. “A Simple Nonparametric Test of Predictive Performance.” *Journal of Business & Economic Statistics* 10 (4): 461–65.

Petropoulos, Fotios, Spyros Makridakis, Vassilios Assimakopoulos, and Konstantinos Nikolopoulos. 2014. "'Horses for Courses' in Demand Forecasting." *European Journal of Operational Research* 237 (1): 152–163.

Pierola, A., I. Epifanio, and S. Alemany. 2016. "An Ensemble of Ordered Logistic Regression and Random Forest for Child Garment Size Matching." *Computers and Industrial Engineering* 101: 455–65. https://doi.org/10.1016/j.cie.2016.10.013.

Priore, Paolo, Borja Ponte, Rafael Rosillo, and David de la Fuente. 2019. "Applying Machine Learning to the Dynamic Selection of Replenishment Policies in Fast-Changing Supply Chain Environments." *International Journal of Production Research* 57 (11): 3663–77. https://doi.org/10.1080/00207543.2018.1552369.

Qu, T., J. H. Zhang, Felix TS Chan, R. S. Srivastava, M. K. Tiwari, and Woo-Yong Park. 2017. "Demand Prediction and Price Optimization for Semi-Luxury Supermarket Segment." *Computers & Industrial Engineering* 113: 91–102.

Shen, Bin, Tsan-Ming Choi, and Stefan Minner. 2019. "A Review on Supply Chain Contracting with Information Considerations: Information Updating and Information Asymmetry." *International Journal of Production Research* 57 (15–16): 4898–4936.

Sillanpää, Ville, and Juuso Liesiö. 2018. "Forecasting Replenishment Orders in Retail: Value of Modelling Low and Intermittent Consumer Demand with Distributions." *International Journal of Production Research* 56 (12): 4168–85. https://doi.org/10.1080/00207543.2018.1431413.

Sugihara, George, and Robert M. May. 1990. "Nonlinear Forecasting as a Way of Distinguishing Chaos from Measurement Error in Time Series." *Nature* 344 (6268): 734.

Svetunkov, Ivan, and John E. Boylan. 2019. "State-Space ARIMA for Supply-Chain Forecasting." *International Journal of Production Research* 0 (0): 1–10. https://doi.org/10.1080/00207543.2019.1600764.

Taskaya-Temizel, Tugba, and Matthew C. Casey. 2005. "A Comparative Study of Autoregressive Neural Network Hybrids." *Neural Networks* 18 (5–6): 781–789.

Terui, Nobuhiko, and Herman K. Van Dijk. 2002. "Combined Forecasts from Linear and Nonlinear Time Series Models." *International Journal of Forecasting* 18 (3): 421–438.

Thomassey, S. 2010. "Sales Forecasts in Clothing Industry: The Key Success Factor of the Supply Chain Management." *International Journal of Production Economics* 128 (2): 470–83. https://doi.org/10.1016/j.ijpe.2010.07.018.

Winkelhaus, Sven, and Eric H. Grosse. 2020. "Logistics 4.0: A Systematic Review towards a New Logistics System." *International Journal of Production Research* 58 (1): 18–43.

Wu, Qi. 2010. "A Hybrid-Forecasting Model Based on Gaussian Support Vector Machine and Chaotic Particle Swarm Optimization." *Expert Systems with Applications* 37 (3): 2388–2394.